

ŽILINSKÁ UNIVERZITA V ŽILINE

**AUTOREFERÁT
DIZERTAČNEJ PRÁCE**

Žilina, apríl, 2020

Mgr. Eduard Vesel

Žilinská univerzita v Žiline
Fakulta riadenia a informatiky

Eduard Vesel, Mgr.

Autoreferát dizertačnej práce

Dynamické vyrovňovanie záťaže v cloudových systémoch s konceptom zeleného počítania

na získanie akademického titulu „**philosophiae doctor**“ (v skratke **PhD.**)
v študijnom programe doktorandského štúdia
informatika

v študijnom odbore:
aplikovaná informatika

Žilina, apríl 2020

Dizertačná práca bola vypracovaná v externej forme doktorandského štúdia na Katedre informatiky, Fakulte riadenia a informatiky Žilinskej univerzity v Žiline

- Predkladateľ:** Mgr. Eduard Vesel
Katedra informatiky
Fakulta riadenia a informatiky
Žilinská univerzita v Žiline
- Školiteľ:** doc. Ing. Jarmila Škrinárová, PhD.
Katedra informatiky
Fakulta prírodných vied
Univerzita Mateja Bela v Banskej Bystrici
- Oponent:** doc. RNDr. PaedDr. Ladislav Huraj, PhD. mimoriadny profesor
Katedra aplikovanej informatiky
Fakulta prírodných vied
Univerzita sv. Cyrila a Metoda v Trnave
- Oponent:** doc. Ing. Ján Hudec, CSc.
Katedra kvantitatívnych metód a informačných systémov
Ekonomická fakulta
Univerzita Mateja Bela v Banskej Bystrici

Autoreferát bol rozoslaný dňa:

Obhajoba dizertačnej práce sa koná dňa 18. 8. 2020 o 14:00 h. pred komisiou pre obhajobu dizertačnej práce schválenu odborovou komisiou v študijnom odbore **informatika**, v študijnom programe **aplikovaná informatika**, vymenovanou dekanom Fakulty riadenia a informatiky Žilinskej univerzity v Žiline dňa

prof. Ing. Karol Matiaško, PhD.
predseda pracovnej skupiny odborovej komisie
študijného programu **aplikovaná informatika**
v študijnom odbore **informatika**
Fakulta riadenia a informatiky
Žilinská univerzita
Univerzitná 8215/1
010 26 Žilina

Úvod

V dnešnej modernej dobe zohrávajú čoraz väčšiu dôležitosť pri vykonávaní náročných úloh aj počítačové technológie ako cloudové systémy, či už vo forme súkromného alebo verejného modelu nasadenia. Využitie cloudového počítania je stále prevažne v komerčnom sektore, ale v poslednej čase sa používa aj na výskum a vedecké výpočty. Zároveň sa dnes pri distribuovaných systémoch stretávame nielen s posúdením vhodnosti použitia virtuálnych alebo fyzických počítačových zdrojov, ale významným faktorom a javom je šetrnosť k životnému prostrediu.

Vo svete informačných technológií a v oblasti vysokovýkonného počítania je etablovaná myšlienka tzv. zeleného počítania. Algoritmy pre efektívne využitie dostupných výpočtových zdrojov sa používajú na dosiahnutie cieľa v rámci vysokovýkonného počítania, ktorý je skrytý pod pojmom zeleného počítania. V spojení s poskytovateľmi služieb v oblasti cloudového počítania je ďalším riešením, aj riešenie zabezpečenia energie dátového centra z obnoviteľných zdrojov.

Používanie virtuálneho prostredia a počítačových zdrojov, ktoré sú k dispozícii na diaľku, môže spôsobiť spomalenie riešených úloh. Na druhej strane, rýchla dostupnosť a cena použitia sú výhodou na overenie a nameranie výsledkov v relatívne primeranom čase.

Preto sme sa rozhodli skúmať práve problematiku a vplyv zeleného počítania v distribuovaných systémoch na vykonávané úlohy. Za týmto účelom sme si zvolili pre otestovanie a určenie vhodných základných spôsobov vyrovnávania pracovnej záťaže simulačné prostredie CloudSim a reálne prostredie cloudu v službe Amazon Web Services.

Začiatok dizertačnej práce uvádza stanovené ciele práce. Práca je rozdelená do 7 kapitol. Kapitola 1 je zameraná na cloudové a výpočtové systémy. Obsahuje definície prostriedkov systémov. Prináša informácie o počítačovom klastri a v súčasnosti najvýkonnejších superpočítačov sveta. V závere kapitoly sa zaoberáme možnosťou vedeckých výpočtov v cloudovom prostredí.

Vyrovňovanie záťaže, rozvrhovanie, migrácia a konsolidácia virtuálnych strojov je v kapitole 2. Tiež tu objasňujeme metódy nastavenia výkonu cloudu.

Tretia kapitola obsahuje metódy používané na znižovanie energetickej náročnosti cloudov – zelené počítanie.

Návrh nových algoritmov na znižovanie energetickej náročnosti je predstavený a opísaný v kapitole 4.

Skúmanie cieľov je obsiahnuté v kapitolách 5 a 6. V týchto kapitolách najprv objasňujeme zvolené metodiky a metódy riešenia pre jednotlivé cloudové prostredia. Ďalej definujeme testovacie úlohy, prípravu zdrojov a nameranie výsledkov.

Vyhodnotenie dosiahnutých výsledkov a budúci rozvoj v danej problematike je zahrnuté v kapitole 7.

Ciele práce

Na základe analýzy teoretických východísk, v oblasti manažovania cloudov, a aktuálneho stavu tejto problematiky doma a v zahraničí vyplýva, že problém zeleného počítania je v súčasnosti jednou z hlavných tém. Táto problematika zahŕňa, v rámci cloudového počítania, optimalizáciu času dokončenia úloh, efektívne a konsolidované virtuálne zdroje a efektívne, metódy znižovania spotreby elektrickej energie cloudových systémov.

V súlade s týmito poznatkami sme stanovili hypotézy:

- H1 - Cloudové systémy, na báze virtuálnych strojov, spomaľujú činnosť fyzických výpočtových prostriedkov, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.
- H2 - Zelené počítanie v cloudových systémoch spomaľuje činnosť počítačového cloudu, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.
- H3 - Dekompozícia cloudovej aplikácie negatívne ovplyvňuje náklady na prevádzku výpočtového systému.

Hlavné ciele pozostávajú z nasledovných krokov:

1. Návrh metód zeleného počítania a ich implementácia v jednotlivých cloudových prostrediach.
2. Výber testovacích úloh.
3. Vykonanie experimentu a overenie funkčnosti metódik vo zvolených prostrediach.
4. Vyhodnotenie nameraných výsledkov.

K dosiahnutiu stanovených cieľov používame simulačné aj reálne prostriedky služieb cloudového počítania.

1 Cloudové a výpočtové systémy

V súčasnosti sa zvyčajne používajú viacjadrové a viacprocesorové počítače. Za posledných niekoľko rokov sa výkon najvýkonnejších počítačov sveta niekoľkonásobne zvýšil. Preto je prirodzeným vývojom tvorba a implementácia paralelných a distribuovaných výpočtov, ale aj manažovanie systémov s prihliadnutím na čo najnižšiu spotrebu elektrickej energie týchto systémov. Hlavným dôvodom k masívnemu používaniu paralelných výpočtov je narastajúca časová zložitosť algoritmov a riešených problémov. Práve viacjadrové procesory umožnili veľký nástup paralelizácie. Úlohy na nich vykonávané, sa spracúvajú na viacerých jadrách, a nie len na niekoľkých jednotlivých procesoroch. Výhodou je, že komunikácia medzi jadrami býva rýchlejšia ako komunikácia medzi procesormi. Je praktickejšie a jednoduchšie použiť výkon takýchto procesorov, aj v počte tisícov, pre dosiahnutie žiadaného cieľového výkonu. [1] [2]

K zbytočnému časovému zdržaniu, ku ktorému prichádza pri náročných sekvenčných programoch, sa dá vyhnúť použitím paralelného programovania a ich výpočtami. Zrýchlenie algoritmu je dosiahnuté vykonávaním čiastkových úloh daného programu súbežne, pri zachovaní správnosti a funkčnosti programu. [3] Paralelný výpočet obsahuje procesy, ktoré môžu byť navzájom od seba nezávislé alebo závislé. Úlohy rozdelené na procesy, ktoré bežia na viacerých procesoroch simultánne, spolu komunikujú pomocou siete alebo zbernice. [4]

1.1 Zdroje a stroje počítačového systému

Zdroj je definovaný ako prvok počítačového systému s limitovanou kapacitou. Limitovaná kapacita vyjadruje napr. počet procesorov, veľkosť pamäte a iné. Zdroje slúžia na

vykonávanie úloh a pracujú s určitou rýchlosťou. Rýchlosť zdroja určuje čas potrebný na vykonanie úlohy na tomto zdroji. Zátťaž zdroja, v priebehu činnosti sa môže meniť, reprezentuje jeho využitie v danom časovom intervale.

Naopak, výpočtový stroj, je množina kumulatívnych zdrojov (procesor, pamäť, úložisko). Opis výpočtového stroja zahŕňa jeho názov, množinu zdrojov a ich architektúru, operačný systém, periférie, atď. Taktiež je to kapacita, zátťaž, rýchlosť a umiestnenie.

Virtuálny stroj je softvérová implementácia výpočtového stroja nad virtualizačnou vrstvou (hypervizorom), ktorý vykonáva operácia tak, ako fyzický stroj.

1.2 Počítačový klaster

Počítačový klaster je jedna logická jednotka pozostávajúca z dvoch alebo viacerých počítačov, ktoré sú navzájom prepojené pomocou lokálnej počítačovej siete, prípadne technológiou infiniband. [5] Sieťovo prepojené počítače sa v podstate prejavujú ako jeden, oveľa výkonnejší stroj.

Najnovší zoznam superpočítačov (z novembra 2019) uvádza, že najvýkonnejším superpočítačom sveta je americký superpočítač s označením Summit, ktorý je umiestnený v DOE/SC/Oak Ridge National Laboratory v meste Oak Ridge. Celkový výkon nameraný v benchmarku Linpack je cez 148 PFlop/s a systém pozostáva z viac ako 2 410 000 jadier, ktoré sú osadené na procesore IBM POWER9. Veľkosť pamäte RAM predstavuje vyše 2,8 PB.

1.3 Gridové a cloudové počítanie

Počítačové klastre môžu byť spolu prepojené do gridov. Počítačový grid je infraštruktúra, ktorá zahŕňa integrované a spolupracujúce použitie počítačov, počítačových sietí, databáz a vedeckých nástrojov vlastnených a manažovaných viacerými organizáciami. [6]

Distribučovaný systém pozostáva zo súboru samostatných nezávislých počítačov, navzájom prepojených prostredníctvom počítačovej siete, ktoré sú vybavené distribuovaným operačným systémom. Pomocou tohto softvéru je umožnené počítačom, aby manažovali a zdieľali zdroje systému, hardvéru a dát.

Rozširovať distribuovaný systém je možné z pohľadu viacerých komponentov: počet používateľov a procesov, maximálna vzdialenosť medzi uzlami a počet domén.

Gridové počítanie je podľa Fostera [7] definované ako „hardvérová a softvérová infraštruktúra, ktorá poskytuje spoľahlivý, konzistentný a lacný prístup k vysokovýkonným výpočtovým kapacitám“ a „využívanie priameho prístupu k počítačom, softvéru, dátam a ostatným zdrojom podľa potreby. Takéto využívanie je nevyhnutne vysoko riadené poskytovateľom zdrojov a používatelia majú jasne definované čo je využívané, kto je oprávnený využívať zdroje a podmienky, za ktorých prichádza k využívaniu.“

Cloudové počítanie sa vzťahuje na poskytovanie prístupu k výpočtovým prostriedkom cez širokopásmovú sieť.

Dohodu o úrovni poskytovaných služieb (angl. Service Level Agreement, SLA) poskytuje prevádzkovateľ infraštruktúry. SLA vyjadruje záväzok poskytnúť určitú kvalitu služieb.

Virtualizácia vo svete počítačov znamená technológiu, pomocou ktorej je možné dosiahnuť vytvorenie virtuálnej verzie reálnych zariadení ako napr. servery, úložné miesta alebo siete. Rovnako je možné vytvoriť virtualizované prostredie (virtualizovanie platforiem), ktorým je celý operačný systém. Týmto spôsobom sa dajú virtualizovať operačné systémy ako Microsoft Windows, Android, Linux, ktoré sú nezávislé od hostiteľského operačného systému a môžu využívať dostupné periférne zariadenia.

Škálovateľnosť počítačových cloudov je daná ich elasticitou. To predstavuje vyrovnanie zaťaženia aplikačných inštancií, ktoré bežia oddelene na rôznych operačných systémoch. Procesory a šírka pásma siete je pridelená a odobraná na požiadanie. Systémové kapacity sa dajú prispôbovať v závislosti od počtu užívateľov, inštancií a množstvo prenesených dát v danom čase.

1.4 Virtuálne stroje

Virtualizácia hardvéru slúži na spúšťanie viacerých operačných systémov a softvéru na jednom fyzickom stroji. Virtualizáciou umožníme vytvorenie viacerých virtuálnych strojov. Každý virtuálny stroj musí obsahovať operačný systém, programy a aplikačný softvér. [8]

1.5 Vedecké výpočty v cloudovom prostredí

K rozhodnutiu počítať paralelné úlohy v cloudovom prostredí treba pristupovať individuálne a zohľadniť aj zameranie úloh. Úlohy obsahujúce intenzívne výpočty majú zvyčajne veľkú réziu na medziprocessorovú komunikáciu a často nie sú dobre škálovateľné. [9] [10]

Z pohľadu celkových finančných nákladov nie je vhodné prenajať si v cloude identickú infraštruktúru, aká sa používa pri fyzických serveroch pre vysokovýkonné počítače. [11] [12] Celkovú výhodnosť cloudovej infraštruktúry ovplyvňuje aj model rezervovania požadovaných inštancií. Oproti modelu pay-as-you-go (priebežný systém platenia) ide o inštalácie na požiadanie, rezervované alebo predplatené dopredu.

2 Vyrovnanie záťaže systému

Záťaž počítača predstavuje množinu úloh, ktorá sa na tomto počítači vykonáva. Záťažou sú predovšetkým úlohy a pracovné toky úloh. Keďže virtuálny stroj je softvérovou implementáciou fyzického stroja, tiež predstavuje záťaž.

Úloha (angl. job, task) je základná jednotka, ktorá sa vykonáva na zdroji. Úloha môže mať špecifické požiadavky na množstvo a rôzne charakteristiky zdrojov, ku ktorým má byť pridelená súvisle alebo na určité časové intervaly, v ktorých ich bude používať. [8]

Stroje môžu pracovať rôznou rýchlosťou, preto čas spracovania úlohy j je potrebné vzťahovať na stroj i .

Vyrovnanie záťaže sa používa na distribúciu a rozloženie väčšieho pracovného zaťaženia uzla na menšie zaťaženie, a tým dosiahnuté zvýšenie celkového výkonu systému. V prostredí cloudového počítania je potrebné dynamicky vyrovnať miestnu záťaž rovnomerne medzi všetkými dostupnými uzly. [13]

Medzi základné spôsoby vyrovňovania pracovnej záťaže v cloudových systémoch patrí rozvrhovanie, migrácia úloh a virtuálnych strojov, replikácie a elasticita. Migrácia úlohy je

definovaná ako prenos úlohy zo súčasného tzv. zdrojového uzla na cieľový miesto a pokračovanie jej vykonávania tam. Môže ísť aj o prenos už rozpracovaných úloh.

Zelené počítanie je možné efektívne implementovať do niekoľkých základných spôsobov vyrovnávania záťaže. Ide o:

- Rozvrhovanie - efektívne pridelovanie nových úloh na zdroje.
- Migráciu virtuálnych strojov.
- Konsolidáciu virtuálnych strojov.
- Metódy nastavenia výkonu cloudu.

2.1 Rozvrhovanie

Rozvrhovanie predstavuje najdôležitejší model ako vyrovnávať záťaž. Rozvrh je priradenie spracovania úloh zdrojom na určitý časový interval tak, že žiadne dve úlohy sa nevykonávajú súčasne na tom istom zdroji a kapacita zdroja nie je prekročená. Rozvrh určuje pre každý časový okamih množinu úloh, ktoré sa v tom okamihu majú vykonávať a množinu zdrojov, na ktorých sa majú vykonávať. Najdôležitejšie kritériá optimálnosti pre rozvrh φ , ktoré treba minimalizovať sú:

- Čas dokončenia poslednej úlohy C_{max} , označovaný tiež ako maximálny čas konca úloh (angl. makespan), $C_{max} = \max_j \{C_1, \dots, C_n\}$ pre rozvrh φ je $C_{max}(\varphi) = \max_{j \in J} \{C_j\}$. Toto kritérium si zasluhuje veľkú pozornosť, predstavuje dobu spracovania celej vstupnej množiny úloh a tým aj dĺžku rozvrhu.
- Celkový čas dokončenia všetkých úloh vypočítame ako súčet časov úloh $\sum_{j=1}^n C_j$ alebo, ak úlohy majú rôzne váhy počítame vážený súčet časov úloh $\sum_{j=1}^n w_j C_j$.

Optimalizovaním týchto kritérií dokážeme výrazne šetriť elektrickú energiu, pretože čím kratšia je doba výpočtu, tým sa spotrebuje menej energie. Optimalizovaním týchto kritérií s kombinácií so systematickou a automatizovanou konsolidáciou a migráciou virtuálnych strojov dosiahneme významný prínos z hľadiska zeleného počítania.

2.2 Migrácia virtuálnych strojov

Pri virtuálnych strojoch sa preferuje, aby neboli závislé na jednom fyzickom hostiteľovi, a v prípade potreby bolo možné s nimi hýbať medzi rôznymi hostiteľmi. Táto prispôsobivosť prináša nové možnosti v zlepšení využívania cloudových aplikácií tým, že sa v systéme nájde vhodné umiestnenie virtuálnych strojov. [14]

Migrácia virtuálneho stroja umožňuje prenos virtuálneho stroja medzi fyzickými uzlami bez prerušenia a s krátkym výpadkom. Migrácia má však negatívny vplyv na výkon aplikácií bežiacich vo virtuálnych strojoch počas migrácie. [15] Migrácia virtuálneho počítača je otázkou milisekúnd, čo šetrí čas, námahu a umožňuje dosahovať plnenie SLA.

Avšak kvôli migráciám virtuálnych strojov alebo ich neoptimalizovanému prideleniu, môžu spolu komunikujúce virtuálne stroje skončiť na rôznych logických jednotkách vzdialených fyzických uzloch. Tie následne poskytujú nákladné dátové prenosy medzi sebou. Ak sú komunikačné virtuálne počítače priradené hostiteľom v rôznych stojanoch, sieťová komunikácia môže zahŕňať ďalšie sieťové prepínače a prepojenia, ktoré spotrebúvajú ďalšie značné množstvo energie. [16]

2.3 Konsolidácia virtuálnych strojov

Z dôvodu dynamického vytvárania a ukončovania virtuálnych strojov v cloudových datacentrách, vzniká na fyzických serveroch fragmentácia zdrojov. Táto fragmentácia vedie k zhoršeniu využívania prostriedkov daného servera. Riešením spomenutých problémov je konsolidácia virtuálnych strojov. Pomocou konsolidácie sa aktívne virtuálne stroje balia na minimálny počet fyzických serverov. Do úvahy sa berie splnenie cieľov, ktorými sú zníženie nákladov, úspora energie a maximalizácia využívania dostupných prostriedkov fyzických serverov.

2.4 Metódy nastavenia výkonu cloudu

V prípade výskytu, alebo ešte pred výskytom výkonnostného problému systému, je potrebné správne spustiť regulačný proces aktívnych zdrojov, tak aby boli splnené kritériá pracovnej záťaže a aplikácií.

Ako uvádza Moghaddam [17], pre zabezpečenie a kontrolu výkonnostných požiadaviek existujú rôzne riešenia zahrňujúce zmeny nastavení zdroja, pridanie nových výpočtových zdrojov (virtuálnych strojov) alebo náhrada existujúcich výpočtových zdrojov za výkonnejšie. Poskytovatelia cloudových služieb prispievajú ku všetkým typom rozhodnutí zohľadňujúcich aplikáciu SLA a dosiahnutia cieľov, akými sú rozpočtové a bezpečnostné obmedzenia počas rozhodovacieho proces.

2.4.1 Metódy identifikácie problematickej aplikácie v cloude

Vlastnosti funkcií nachádzajúcich sa v tejto kategórii nie sú priamo špecifické len pre cloudové systémy, pretože sú aplikované priamo do aplikácií a ich prostredia. Na degradáciu výkonu operačného systému a aplikácií v dôsledku interných problémov, ktorými môžu byť poškodenie údajov, vyčerpanie zdrojov operačného systému alebo akumulácia číselných chýb sú zamerané metódy aplikačnej úrovne. [18]

Pokus o identifikáciu problematickej aplikácie alebo systémového komponentu, vyčistenie interného stavu alebo reštartovanie komponentu je softvérovou možnosťou pri týchto typoch problémov.

2.4.2 Nadmerné rezervy

Celkový výkon systému a kvalita služieb ponúkaných aplikácií ovplyvňuje riadny odhad požadovaných zdrojov, a efektívne využitie ponúkaných cloudových zdrojov. Špeciálne pri systémoch s dynamickou záťažou a časovo závislými výkyvmi je odhad zdrojov komplexným problémom. Na vyriešenie novej nestability systému a zvládnutie zdrojových požiadaviek aplikácií, je poskytnutie statického dostatku zdrojov, ktoré majú zvládať predpokladané maximálnu záťaž systému. Nevýhodou tohto riešenia z pohľadu poskytovateľov cloudových služieb, celkových nákladov a energetickej náročnosti je, že vrcholy pracovnej záťaže sú zvyčajne len dočasnými udalosťami, ktoré sa vyskytujú zriedka a netrávajú dlho. To znamená, že väčšinu času sú alokované zdroje nevyužitú.

2.4.3 Metódy automatického škálovania

Škálovateľnosť je schopnosť systému, aplikácie alebo virtuálneho stroja riadiť zmenené požiadavky na zdroje. Toto je jedna z najcennejších a prevažujúcich funkcií cloudového počítania. Vďaka škálovateľnosti môžeme v rámci virtuálneho stroja zväčšiť diskovú kapacitu

pre ukládanie údajov, veľkosť operačnej pamäte RAM alebo počet dostupných procesorov, aby vyhovovali dynamicky sa meniacim požiadavkám a SLA kritériám. Z pohľadu systému môžeme meniť celkový počet virtuálnych strojov.

3 V Zelené počítanie – metódy používané na znižovanie energetickej náročnosti cloudov

Existuje niekoľko techník a metód, založených na fyzickom alebo softvérovom princípe, ktoré prispievajú k znižovaniu energetickej náročnosti systémov tým, že:

- Definujú modely spotreby elektrickej energie v cloudoch.
- Detegujú preťažené uzly.
- Predpovedajú zaťaženie uzlov.

3.1 Pravidlo statického prahu

Medzi najjednoduchší detekčný algoritmus preťaženia uzla sa považuje pravidlo statického prahu. [19] Uzol sa považuje za preťažený, ak procesorové využitie aktuálneho uzla prekročí špecifikovaný prah využitia.

3.2 Dynamické škálovanie napätia a frekvencie

Spotreba energie výpočtových uzlov v dátových centrách je v zásade vyjadrená spotrebou procesora (CPU), diskovým úložiskom a sieťovým rozhraním. V porovnaní s ostatnými systémovými zdrojmi, značnú časť z tejto energie spotrebovávajú CPU. Okrem toho, využitie CPU je často úmerné celkovému systémovému zaťaženiu. Riadenie spotreby energie a efektívne využitie je možné ovplyvniť technikou dynamického škálovanie napätia a frekvencie (angl. dynamic voltage and frequency scaling, DVFS).

Beloglazov a kol. [20] poukázali, že aplikovanie techniky DVFS na CPU končí v takmer lineárnom vzťahu spotreba-frekvencia pre server (so zvýšenou frekvenciou súvisí zvýšená spotreba, a naopak). Je to výsledkom toho, že DVFS je aplikovaný len na CPU a nie ostatné systémové komponenty.

Štúdie tiež preukázali, že server priemerne v stave nečinnosti spotrebúva o 70% elektrickej energie menej ako keď pracuje na plnom výkone CPU.

Na používanie rôznych techník aplikovania zeleného počítania a ich dopadov na výkonnosť cloudových dátových centier sa venovali aj Bala et al. [21]. Tí v prostredí GreenCloud Simulator ukázali, že aj v dnešných tradičných datacentrách je možné zredukovať spotrebu energií a tak znížiť uhlíkovú stopu. Kľúčom k dosiahnutiu tohto výsledku bolo aplikovanie režimu správy napájania založeného na technikách DVFS a Dynamic Network Shutdown.

3.3 Metóda medzikvartilového rozpätia

Štatistickou metódou na zistenie preťaženie hostiteľa je medzikvartilové rozpätie (IQR) stanovením hornej hranice využitia. [22] IQR je tiež nazývané stredný rozptyl alebo polovica. Autori vypočítali IQR z rozdielu medzi prvým (Q1) a tretím (Q3) kvartilom, kde Q1 je prvých 25% a Q3 je tretích 25% z vybranej množiny.

3.4 Metóda lokálnej regresie

Jednou z metód na detekciu preťaženého uzla (hostiteľa) je metóda lokálnej regresie. Výsledkom kombinácie lokálnej regresie a medzikvartilového rozpätia je metóda lokálnej robustnej regresie. [22]

Hlavnou myšlienkou metódy lokálnej regresie je vsadenie jednoduchých modelov do lokalizovaných podmnožín údajov pre vytvorenie krivky, ktorá sa blíži pôvodným údajom.

Tieto metódy sú použité na predpovedanie ďalšieho využitia CPU na základe štatistickej analýzy historických údajov využitia vybraného hostiteľa v cloudovom dátovom centre.

3.5 Model mediánovej absolútnej odchýlky

Mediánová absolútna odchýlka, je štatistický model používaný na detekciu preťaženého hostiteľa, určením vrchného a spodného prahu pre všetky virtuálne stroje spustených na tomto hostiteľovi.

Model počíta predpoveď zaťaženia pre hostiteľa. Hostiteľ sa považuje za preťažený, pokiaľ predpokladaná hodnota zaťaženia presahuje určený vrchný prah. V takom prípade by niektoré virtuálne stroje mali byť migrované na iného hostiteľa za účelom zredukovania zaťaženia, aby sa predišlo prípadným porušeniam SLA.

4 Návrh nových algoritmov na znižovanie energetickej náročnosti cloudov

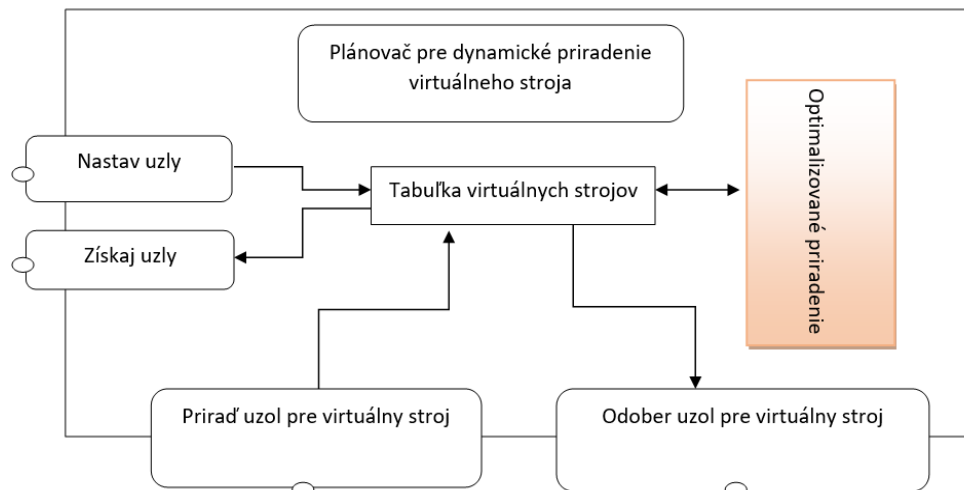
Týmito algoritmi sú:

- algoritmus energetickej efektívnosti - dynamické plánovanie (AEEDP),
- algoritmus energetickej efektívnosti – fuzzy logika (AEEFL).

4.1 Algoritmus energetickej efektívnosti – dynamické plánovanie

Prvým z predstavených vlastných navrhnutých algoritmov plánovania pre riešenie znižovania energetickej náročnosti v rámci cloudových systémov je algoritmus energetickej efektívnosti - dynamické plánovanie (AEEDP) (Obrázok 4.1).

Algoritmus je navrhnutý tak, aby sa dynamicky vykonávala migrácia virtuálnych strojov na uzle s dostupnými prostriedkami s cieľom ich neustáleho zlepšovania umiestnenia.



Obrázok 4.1. Navrhnutý plánovač pre dynamické priradenie virtuálneho stroja

Do algoritmu sme implementovali funkciu migrácie Best Fit pre získaný zoznam virtuálnych strojov a zaťaženia jednotlivých uzlov. Ten má za cieľ informovať, ktoré virtuálne stroje migrovať. Vrátený zoznam predstavuje postupnosť migrácií, ktoré sa majú vykonať. Každá položka v zozname obsahuje identifikátor virtuálneho stroja a identifikátor cieľového hostiteľa.

Z tohto dôvodu výsledná simulácia dátového centra spotrebováva menej energie ako pri simulácii bez aplikácii žiadnej metódy zeleného počítania.

4.2 Algoritmus energetickej efektívnosti – fuzzy logika

Druhým, hlavným navrhnutým algoritmom, je algoritmus energetickej efektívnosti – fuzzy logika (AEEFL).

4.2.1 Opis pojmov a vzťahov fuzzy logiky

Fuzzy inferenčné systémy (FIS) sú určené na vytváranie výstupov a vychádzajú z konceptov fuzzy logiky. Jeden z takých bol navrhnutý L. A. Zadehom, v jeho práci s fuzzy množinami. [23]

Fuzzy logika je forma logiky, v ktorej sa hodnoty pre pravdu pohybujú od nuly po jednu. Fuzzy logika je založená na ľudských vágnych rozhodnutiach a pomocou matematiky vyjadrujúca neurčitosť. Je určená na prácu s nepresnými informáciami pomocou teórie fuzzy množín, transformácií, fungovania a vyvodenia záverov. Vo všeobecnosti možno spracovanie fuzzy logiky opísať niekoľkými fázami:

- Fuzzifikácia – je proces transformácie „fuzzifikačných“ vstupov reálnych údajov do triedy funkcií príslušnosti.
- Inferencia – je proces spojenia vstupov fuzzy systému s jeho pravidlami na výpočet hodnôt fuzzy výstupných premenných (výstupných funkcií).
- Defuzzifikácia – je proces transformácie fuzzy výstupných funkcií na ostré hodnoty.

Pre transformáciu a spracovanie reálnych údajov do tvaru fuzzy množín so stupňom príslušnosti zahŕňa fuzzy logika výraz jazykové premenné. Ten je určený na reprezentovanie skutočných vstupných údajov v zmysle jazykových premenných, ktoré sa vyznačujú jedným alebo viacerými jazykovými označeniami. Každé jazykové označenie má svoju vlastnú funkciu príslušnosti a základný priestor všetkých prvkov (tzv. univerzum) jazykovej premennej. Jazykové premenné a hodnoty sa vzťahujú na skutočné ľudské skúsenosti používajúc prirodzený jazyk na vyjadrenie jednej alebo viacerých udalostí, pričom prvá je iba definíciou akéhokoľvek skutočného procesu a druhá je podobná matematickému nahradeniu.

Systémy fuzzy logiky zahrňujúce fázy fuzzifikácie, spracovania pravidiel a defuzzifikácie sa nazývajú fuzzy inferenčný systém alebo fuzzy regulátor. Jedným z najbežnejších je fuzzy model typu Mamdani, ktorý navrhol E. H. Mamdani [24] a je rozšírený medzi širokou vedeckou komunitou.

Pre vytvorenie výstupnej premennej fuzzy funkcie, pozostáva inferenčná fáza z tzv. krokov agregácie, aktivácie, akumulácie a zaobraním sa bázou pravidiel.

Existuje mnoho rôznych metód defuzzifikácie, pričom niektoré z nich sú:

- Metóda ťažiska (MT).
- Metóda stupňov

- Metóda najvýznamnejšieho maxima (výsledok môže byť najviac vľavo, v strede, najviac vpravo).

V našom algoritme je pre defuzzifikáciu použitá metóda ťažiska. To znamená výpočet ťažiska pre výstupnú funkciu ako (4.1),

$$\forall x \in X: MT = \frac{\int x\mu_O(x)dx}{\int \mu_O(x)dx} \quad (4.1.)$$

kde O je výstupná fuzzy množina a $X=\{x\}$ sú všetky prvky univerza.

4.2.2 Detailný opis algoritmu pre detekciu preťaženia uzla

Existuje niekoľko techník detekcie preťaženia uzla, kým sa vykonávajú úlohy konsolidácie virtuálnych strojov. Takými to z nich sú mediánová absolútna odchýlka (MAD), medzikvartilové rozpätie (IQR), regresné stratégie a iné. [22]

Všetky vyššie uvedené techniky aplikujú ako vstup históriu využitia uzla a vytvárajú prahy (tzv. thresholds) pre MAD a IQR, a predpokladané využitie uzla pre regresné metódy. V princípe môžu byť výstupy techník MAD, IQR a regresia zlúčené, aby sme tak získali všetky výhody z každej metódy. Za týmto účelom sme použili fuzzy logiku, ktorá sa často používa v prípade, keď je ťažké (alebo dokonca nemožné) vytvoriť priamy matematický alebo fyzikálny model kvôli neistote v dátach. Fuzzy logika zahŕňa FIS. FIS obsahuje vstupné premenné, výstupné premenné, hodnoty príslušnosti a pravidiel. Výsledkom sú vytvorené výstupné hodnoty výstupných premenných na základe vstupných hodnôt a navrhnutých pravidiel.

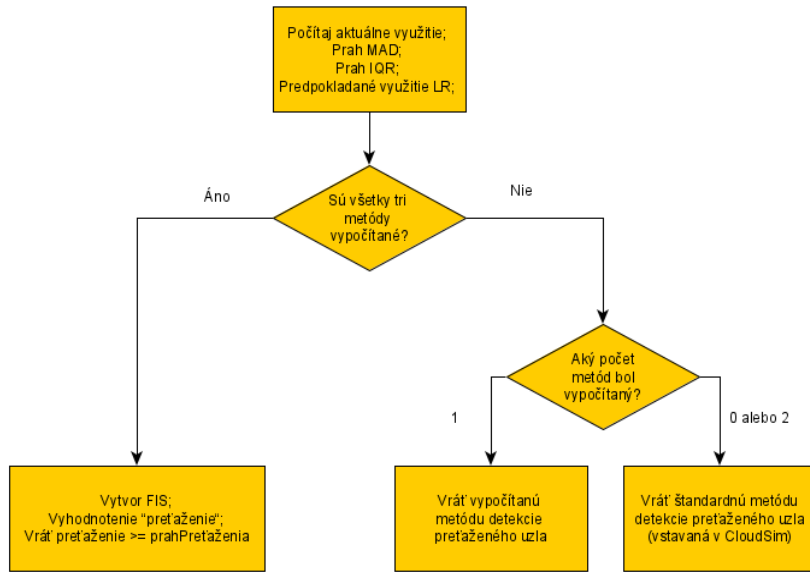
V našom navrhnutom algoritme sme nastavili MAD, IQR a lokálnu regresiu (LR) ako vstupné hodnoty, a ako výstupnú hodnotu považujeme premennú preťaženie (t. j. preťaženie uzla), nachádzajúcu sa v rozsahu $\{0,1\}$, ktorá zodpovedá podmienkam pravdepodobnosti preťaženia uzla.

Na vytvorenie pravdepodobnosti preťaženia uzla sme použili lichobežníkové funkcie príslušnosti a stanovených 27 pravidiel. FIS sa pre zistenie preťaženia uzla zakaždým generuje dynamicky, pretože prahy pre MAD, IQR a predpokladané využitie uzla sú v priebehu času rozdielne.

Blokový diagram na detekciu preťaženia uzla je znázornený na obrázku 4.2. Na začiatku je potrebné vypočítať a získať hodnoty pre aktuálne využitie uzlov, prahy MAD a IQR, a predpokladané využitie pomocou LR. V prípade, že sú všetky tri metódy vypočítané a k dispozícii, prechádzame do stavu vytvárania FIS, vyhodnotenia výstupu FIS, aby sme sa rozhodli, či je hosťiteľ nadmerne preťažený alebo nie (na základe porovnania výstupu FIS s premennou prahPreťaženia). V opačnom prípade, keď nemáme všetky 3 metódy vypočítané, prechádzame k rozhodovaciemu stavu Aký počet metód bol vypočítaný? . Môžu existovať prípady (vyskytujúce sa veľmi zriedka), keď nastane chyba (napríklad nezaradená história využitia), a až žiadna z metód nemusí byť vypočítaná.

Podľa počtu metód, ktoré máme k dispozícii v tomto bode, nastávajú nasledovné varianty:

- 1 metóda - ak vypočítame iba jednu metódu (MAD, LR alebo IQR), použijeme túto metódu na vyhodnotenie preťaženia uzla.
- 0 alebo 2 metódy - ak máme nula alebo práve dve metódy (MAD a LR, MAD a IQR alebo IQR a LR), použijeme preddefinovanú metódu (politiku) „núdzového“

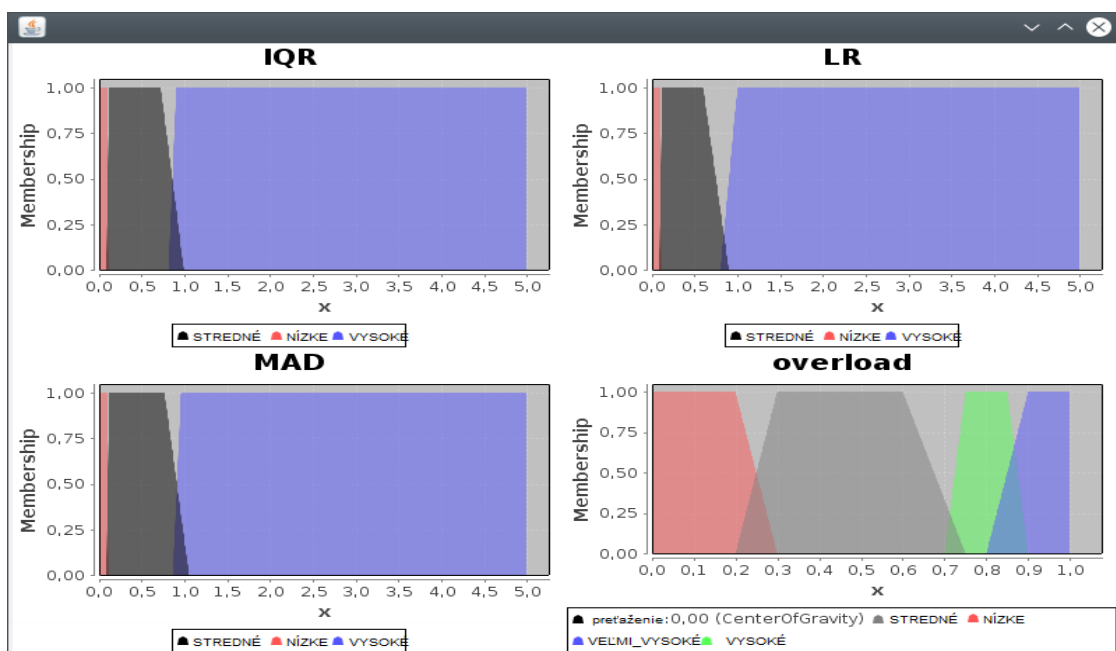


Obrázok 4.2. Blokový diagram

K fuzzifikácii vstupov používame lichobežníkovú funkciu príslušnosti. Fuzzifikačná fáza zahŕňa výpočty stupňa príslušnosti pre konkrétne jazykové premenné. Každá jazyková premenná má svoje vlastné funkcie príslušnosti. V našom prípade sú to vždy lichobežníkové funkcie príslušnosti.

Všetky premenné, vstupy aj výstupy, majú svoje jazykové premenné. MAD, IQR a LR majú jazykové premenné NÍZKE (LOW), STREDNÉ (MEDIUM) a VYSOKÉ (HIGH). Rovnaké pojmy NÍZKE, STREDNÉ, VYSOKÉ a nový pojem VELMI_VYSOKÉ (VERY_HIGHT), sú definované pre premennú preťaženie.

Vytvorený FIS hodnôt príslušnosti je zobrazený obrázku nižšie.



Obrázok 4.3. Vytvorený FIS hodnôt príslušnosti (vygenerovaný)

Obrázok 4.3 znázorňuje vykreslené funkcie príslušnosti pre premenné FIS IQR, LR, MAD a preťaženie. Zároveň sú tu graficky zobrazené všetky hodnoty pre jazykové premenné „NÍZKE“, „STREDNÉ“, „VYSOKÉ“ a „VELMI_VYSOKÉ“ pre vstupné a výstupné premenné. Z grafického zobrazenia vidíme tvary jednotlivých funkcií. Os X znamená konkrétne hodnoty, os Y (príslušnosť) znamená stupeň príslušnosti jazykových premenných pre klasické vstupné alebo výstupné premenné.

Ďalej zabezpečujeme do existujúceho FIS pre MAD, IQR prahy aktuálneho využitia a pre LR – predpokladané LR využitie. Potom voláme funkciu vyhodnotenie() k vytvoreniu FIS výstupu z našich predchádzajúcich troch vstupov: MAD, IQR a LR (Algoritmus 4.14).

Inak povedaná, vyhodnotenie() počíta stupeň príslušnosti, následne aplikujeme inferenčnú fázu a nakoniec vytvárame konkrétne hodnoty premenných preťaženia uzla.

FIS vytvára premennú preťaženie v rozsahu $\{0,1\}$. Nakoniec prichádza k rozhodnutiu, či je tento hosťiteľ preťažený alebo nie, na základe porovnania výstupu FIS s premennou prahPreťaženie.

5 Metodiky a experimentálne overovanie navrhnutých algoritmov

Táto kapitola obsahuje modely riešenia, postupnosť metód, nami navrhnutú metodiku práce a experimentálne overenie nami navrhnutých algoritmov pre dosiahnutie stanovených cieľov práce v simulačnom prostredí.

5.1 Metodika overovania riešenia v simulačnom prostredí CloudSim

Základom pre získanie výsledkov simulácie je v prvom rade vytvorenie simulovaného dátového centra, na ktorého uzloch budú umiestňované virtuálne stroje. V simulácii budeme pracovať s jedným datacentrom, v ktorom bude celkovo k dispozícii 800 uzlov (heterogénneho typu, rozdelených na dve rovnako veľké skupiny) a k nim v priebehu času priradených 1052 virtuálnych strojov s vopred definovanou rozdielnou pracovnou záťažou.

Počas simulácie je celkovým sledovaným obdobím 1 deň, resp. 86 400 sekúnd. Zároveň v simulácii používame 4 rozdielne konfigurácie pre virtuálne stroje.

Pre porovnanie a vyhodnotenie výsledkov aplikovaného konceptu zeleného počítania používame 6 techník:

- statický prah (SP) [15, 22]
- dynamické škálovanie napätia a frekvencie (DVFS) [15, 22]
- lokálna regresia (LR) [15, 22]
- mediánová absolútna odchýlka (MAD) [15, 22]
- algoritmus energetickej efektívnosti – dynamické plánovanie (AEEDP)
- algoritmus energetickej efektívnosti – fuzzy logika (AEEFL)

Zvolené techniky zeleného počítača a navrhnuté algoritmy spolu s vybranou testovacou pracovnou záťažou implementujeme do vytvoreného simulovaného dátového centra. Počas overovania účinnosti zvolených techník sme museli zvoliť aj najvhodnejší prístup pre výber virtuálneho stroja, ktorý má byť migrovaný. Sériu meraní sme porovnali s prístupmi aplikujúcich maximálnu koreláciu, minimálny čas migrácie a náhodný výber. Prístup maximálnej korelácie vyšiel s najlepšimi výsledkami spomedzi ostatných, a preto len ten uvádzame ďalej v práci.

5.2 Simulačné prostredie CloudSim

CloudSim podporuje poskytovanie zdrojov na dvoch úrovniach: na úrovni uzla a na úrovni virtuálnych strojov. Na úrovni uzla môžeme určiť, koľko z celkového výkonu bude spracované a pridelené každému virtuálnemu stroju. Na úrovni virtuálneho stroja dostávajú a používajú aplikácie časť výpočtového výkonu priradená virtuálnemu stroju. Medzi podporovanými distribučnými modelmi cloudového počítania v CloudSim sú zahrnuté IaaS, PaaS a SaaS.

5.2.1 Testovacie úlohy pre simulačné prostredie CloudSim

Pre naše testovanie sme vybrali množinu údajov, ktoré sú používané ako cloudlety pre simuláciu. Množina údajov pozostáva z hodnôt záťaže procesora z reálneho počítačového systému (uzla). Podľa počtu cloudletov sa v priebehu simulácie vytvára rovnaký počet virtuálnych strojov. Každý cloudlet obsahuje informácie o záťaži, ktorú bude predstavovať virtuálny stroj na priradenom uzle. Celkovo máme k dispozícii informácie o 1052 reálnych počítačových systémoch.

5.2.2 Príprava zdrojov

Počet a vlastnosti vytváraných jednotlivých virtuálnych strojov, sú definované pomocou premenných `public final static int`, ktoré určujú:

- VM_TYPES – počet virtuálnych strojov.
- VM_MIPS – výkon virtuálneho stroja.
- VM_PES – počet prvkov spracovania.
- VM_RAM – veľkosť operačnej pamäte.
- VM_BW – rýchlosť siete.
- VM_SIZE – veľkosť virtuálneho stroja.

Počet a vlastnosti vytváraných jednotlivých uzlov sú definované rovnakým spôsobom ako v predchádzajúcom prípade virtuálne stroje. Rozdielom je len prefix namiesto VM -> HOST a SIZE -> STORAGE.

5.3 Experimentálne overovania navrhnutých algoritmov

Tieto experimenty navrhujeme s cieľom potvrdiť alebo vyvrátiť hypotézu H2 - Zelené počítanie v cloudových systémoch spomaľuje činnosť počítačového cloudu, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.

Možné zníženie energetickej náročnosti, oproti simulácii bez optimalizácie, je dosiahnuté konsolidáciou virtuálnych strojov, uvoľnením uzla a tým jeho neaktivitu. V tomto stave je možné uzol hibernovať a tým výrazne znížiť celkovú spotrebu energie. Aj v tomto prípade vyšla technika AEEFL zo všetkých testovaných najlepšie, keď sme zaznamenali zníženie energetickej náročnosti oproti bez optimalizácie až o 2559% (25 násobne menšiu). V porovnaní s inými technikami zeleného počítania, bol najmenší rozdiel medzi SP a AEEFL, a to konkrétne o 69,64% v prospech AEEFL.

Počet vykonaných migrácií virtuálnych strojov počas simulácie vytvoreného cloudového prostredia pre navrhnuté algoritmy a jednotlivé techniky zeleného počítania s kritériom výberu virtuálneho stroja algoritmu maximálnej korelácie, sú uvedené v Tabuľke 5.2. Z

výsledkov v tabuľke vidíme, že náš algoritmus používajúci fuzzy logiku sa javí najlepšie. Počet migrácií je pri tejto technike až o 88,4% nižší ako pri najlepšom výsledku zo známych techník LR a až o 93.3% nižší v porovnaní s technikou SP.

Menší počet migrácií virtuálnych strojov predstavuje efektívnu konsolidáciu, dodržanie celkového kritéria SLA a v konečnom dôsledku aj menšie množstvo sieťového zaťaženia.

Tabuľka 5.1. Energetická náročnosť dátového centra v kWh za sledované obdobie (najnižšia spotreba zvýraznená tučným)

Typ úlohy	Bez optimalizácie	SP	DVFS	LR	MAD	AEEDP	AEFL
workload.20110303	2410,80	153,76	803,91	150,33	176,13	289,14	90,64

Tabuľka 5.2. Počet vykonaných migrácií virtuálnych strojov za sledované obdobie

Typ úlohy	Bez optimalizácie	SP	DVFS	LR	MAD	AEEDP	AEFL
workload.20110303	0	40187	0	23004	23691	6781	2666

Na základe analýzy získaných výsledkov vykonaných simulácií a meraní, môžeme vyhodnotiť vplyv jednotlivých metód a algoritmov na stanovené kritéria. Výsledný rozdiel v celkovej elektrickej spotrebe simulovaného dátového centra počas doby 1 deň je medzi najhoršou technikou (bez optimalizácie), a najlepšou (algoritmus energetickej efektívnosti – fuzzy logika) viac ako 25 násobný.

Experimenty overenia riešenia v simulačnom prostredí CloudSim ukázali, že hypotéza H2 je vyvrátená. To znamená, že neplatí, že zelené počítanie v cloudových systémoch spomaľuje činnosť počítačového cloudu, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.

Tabuľka 5.3. Prehľad porušení SLA

Merania	Bez optimalizácie	SP	DVFS	LR	MAD	AEEDP	AEFL
Porušenie SLA (v %)	0	0,01619	0	0,00677	0,00739	0,01518	0,00594

Aplikovaním techník zeleného počítania sa dosiahla výrazná úspora elektrickej energie v rámci simulovaného dátového centra a miera porušenia SLA vzrástla len minimálne, oproti neaplikovaniu techniky bez optimalizácie (pozn. pozri Tabuľka 5.3). V tomto smere sa meraniami najlepšie ukázal náš algoritmus používajúci fuzzy logiku pre detekciu preťaženia uzla. Hodnota porušenia SLA, počas doby sledovanej simulácie 1 deň na zvolenej testovacej úlohe, predstavovala akceptovanú hodnotu iba 0,00594%.

6 Metódy riešenia aplikácií v cloudových systémoch s cieľom dosiahnuť zelené počítanie

Táto kapitola obsahuje modely riešenia, postupnosť metód, nami navrhnutú metodiku práce a experimentálne overenie úloh v reálnom prostredí.

6.1 Metodika overovania riešenia v reálnom cloudovom systéme

V ponuke dostupného verejného cloudu, vhodného na vykonanie vysokovýkonného počítania, existuje niekoľko poskytovateľov služieb – Amazon, IBM, Microsoft Azure, Google Cloud atď. Pre výskum v reálnom cloudovom prostredí sme zvolili službu Amazon Web Services, konkrétne Amazon EC2, od spoločnosti Amazon. Modelom nasadenia je infraštruktúra ako Služba.

V prostredí reálneho cloudu sme použili 3 typy testovacích úloh.

6.1.1 Testovacie úlohy pre reálne prostredie cloudového systému

V reálnom prostredí cloudového systému sme sa rozhodli použiť tri typy testovacích úloh. Prvá úloha, využíva programovací model rozhrania na výmenu správ (MPI) s názvom Cloverleaf. [25]

Druhý typ úlohy [26], predstavuje paralelný program na ofarbovanie hrán grafov, ktorý obsahuje rôzne nadväzujúce a neprerušiteľné podúlohy. Úloha používa sadu grafov z databázy House of Graphs, ktorá pozostáva z 19 935 34-vrcholových snarkov

Úlohy tretieho typu sú tiež sady grafov z databázy House of Graphs [27]. V tomto prípade sme overovali nasledovný počet grafov pre snarky s 34 vrcholmi.

6.2 Reálne prostredie cloudového systému

Možnosti ponúkané zvolenou službou Amazon EC2 spĺňajú naše požiadavky pre stanovené kritéria jednorazového testovania úlohy v rámci vysokovýkonného počítania. Týmito kritériami sú výber dostupného regiónu datacentra, dostupných výpočtových zdrojov a nástroja na manažovanie vytvoreného klastra

6.2.1 Príprava zdrojov

Z ponuky všetkých dostupných konfigurácií virtuálnych strojov [13] je pri overovaní riešenia použitý typ inštancie optimalizovaný na intenzívne výpočty c4.4xlarge.

6.2.2 Správca klastra

Aby sme mohli jednoducho, rýchlo a automatizovane vytvárať resp. odstraňovať vysokovýkonné klastre, v rámci služby AWS, použili sme voľne dostupný softvér správcu klastra AWS ParallelCluster. [28] AWS ParallelCluster sa inštaluje v prostredí operačného systému na lokálnom počítači. Prvotná konfigurácia po inštalácii prebieha pomocou nastavení account credentials a security key patriacim k vytvorenému účtu v AWS

6.3 Experimentálne overovanie úlohy CloverLeaf

Po vytvorení zadaného klastra, prihlásení sa na nod je potrebné vytvoriť súbor so spúšťacími parametrami a kompiláciou.

Z nameraných východiskových hodnôt bez použitia optimalizácie je možnosťou ich zlepšenia použitie optimalizačných techník pri plánovači Torque. [8]

6.4 Experimentálne overovanie úlohy ofarbovania grafov

Táto časť práce slúži na potvrdenie, príp. vyvrátenie hypotézy H1 - Cloudové systémy, na báze virtuálnych strojov, spomaľujú činnosť fyzických výpočtových prostriedkov, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme. Rovnako tiež skúmame potvrdenie, príp. vyvrátenie hypotézy H3 - Dekompozícia cloudovej aplikácie negatívne ovplyvňuje náklady na prevádzku výpočtového systému.

Pri druhej zvolenej úlohe sa venujeme správaniu úlohy a jej porovnaniu fyzického počítačového klastra a využitím dostupných výpočtových systémov s virtualizáciou na báze virtuálnych strojov. Hlavným cieľom skúmania druhej úlohy je vplyv virtualizácie na algoritmy obsahujúce rekurzívne funkcie.

Pri malých úlohách (100, 1 000 a 1 000 000 grafov), sa na základe celkových časov výpočtu preukazoval systém s virtualizáciou podstatne lepší ako systém s fyzickými strojmi. S narastajúcou veľkosťou ofarbovanej úlohy sa ale zvolené výpočtové systémy začali výkonnostne približovať.

Akonáhle sa výpočtová zložitosť (resp. náročnosť) úlohy zvyšuje, výkon medzi fyzickým počítačovým systémom a virtuálnym klastrom sa začína vyrovnávať. V tomto hraničnom prípade už ide o ofarbovanie veľkej množiny grafov – cca 10 miliónov. Pričom sa dá predpokladať, že pri ďalšom zväčšovaní množiny grafov sa čas výpočtu na cloude s virtuálnymi strojmi podstatne zhorší.

Na základe nameraných hodnôt konštatujeme, že pokiaľ nejde o mimoriadne veľké výpočtové úlohy, môžeme vyvrátiť hypotézu H1 - Cloudové systémy, na báze virtuálnych strojov, spomaľujú činnosť fyzických výpočtových prostriedkov, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.

Pri vyhodnocovaní a skúmaní hypotézy H3 sme použili tri rôzne veľkosti sady grafov s 34 vrcholmi. Kritériom pre vyvodenie záveru je porovnanie dosiahnutých časov výpočtu ofarbovania grafov s použitím dekomponovaných podúloh, Porovnávanie robíme znova medzi výpočtami na fyzickom výpočtovom klastri a cloudovom klastri s virtuálnymi strojmi. Znova, doktorand Mgr. Adam Dudáš poskytol programy a namerané výsledky časov ofarbovania grafov, ktoré robil na fyzickom výpočtovom klastri, bez použitia virtuálnych strojov.

Pri úlohách o veľkosti množiny grafov 3 833 587 a 25 286 953 sme pozorovali výrazne rozdiely medzi jednotlivými typmi klastrov, a to z pohľadu úspešnosti dokončenia úlohy. Ani pri experimentálnej testovacej zostave virtuálneho klastra s dostupnou operačnou pamäťou 768GB RAM, nebola úloha pri použití jednotného modelu dekompozície úspešne dokončená. Zvyšujúca sa pamäťová náročnosť uvedených úloh, s pôvodným modelom dekompozície, spôsobovala zlyhanie vykonania vo virtuálnom prostredí.

Z tohto dôvodu sme museli obe zvyšné množiny grafov dekomponovať na maximálne rovnako veľké podúlohy.

Tabuľka 6.1. Namerané hodnoty pre množinu 25 286 953 grafov

Názov	Fyzický počítačový klastor ¹	Virtuálny klastor
Interval dokončenia úlohy (celkový čas + CPU čas výpočtu)	04:19:09 hodín	04:49:35 hodín

¹ Výsledky poskytnuté doktorandom Mgr. Adamom Dudášom

Uvažujme, že najväčší vplyv na znižovanie energetickej náročnosti pri prevádzke výpočtových systémov, z pohľadu aplikácií, má znižovanie času výpočtu jednotlivých úloh. Na základe skúmania úloh s 3,8 miliónmi grafov a 25 miliónmi grafov s dekompozíciami úlohy na rovnako veľké výpočtové podúlohy (pričom je zrejmé, že rôzne architektúry si vyžadujú rôzne dávky úloh) môžeme konštatovať, že zamietame hypotézu H3 – Dekompozícia cloudovej aplikácie negatívne ovplyvňuje náklady na prevádzku výpočtového systému.

7 Vyhodnotenie a pokračovanie výskumu

Záver práce, a táto kapitola, je venovaná vyhodnoteniu nameraných výsledkov pomocou navrhnutých algoritmov a zvolenej metodiky overovania.

V prvých troch kapitolách, tvoriacich hlavnú teoretickú časť práce, sme objasnili aktuálnu situáciu v oblasti počítačových a distribuovaných systémov. Ďalej, to bola aktuálna situácia v oblasti vyrovnávania záťaže systému a metód používaných pri znižovaní energetickej náročnosti cloudov.

V kapitole 4 sme predstavili návrh dvoch nových algoritmov zameraných na znižovanie energetickej náročnosti cloudov. Oba algoritmy vychádzali z rozdielneho prístupu ako dosiahnuť stanovené ciele. Prvý, algoritmus energetickej efektívnosti – dynamické plánovanie, vykonáva neustále migrovanie virtuálnych strojov na základe zoznamu uzlov a ich zaťaženia. Práve zlepšovanie umiestňovania virtuálnych strojov a znižovanie záťaže uzlov spôsobilo v konečnom dôsledku zníženie energetickej náročnosti. Druhým algoritmom je algoritmus energetickej efektívnosti - fuzzy logika. Fuzzy logika je použitá pre detekciu preťaženého uzla, čo má za následok konsolidáciu virtuálnych strojov a jeho uvoľnenie. Vyhodnotenie, či je uzol preťažený, vyplýva z výstupnej premennej fuzzy inferenčného systému a bázy dvadsiatich siedmich pravidiel.

Dosiahnuté výsledky a namerané časy z experimentálneho overovania v jednotlivých cloudových prostrediach sme opísali v kapitolách 5 a 6. Po získaní výsledkov aplikovaním vybraných optimalizačných metód v simulačnom prostredí CloudSim, vidíme rozdiel v celkovej spotrebe elektrickej energie medzi jednotlivými technikami. Príjemným zistením je, že na vybranej testovanej úlohy získal najlepšie výsledky algoritmus, ktorý sme navrhli. Výsledok sme dosiahli pri akceptovateľnom a minimálnom porušení SLA, na základe dobre pripraveného vyhodnocovacieho systému pravidiel v rámci inferenčného systému.

Pri druhom zvolenom cloudovom prostredí, ktorým bol reálny cloudový systém, boli experimenty zamerané na vyhodnotenie hypotéz H1 a H3. Obe hypotézy boli vyvrátené, vychádzajúc z porovnania nameraných časov dokončenia úloh medzi fyzickým počítačovým klastrom a virtuálnym klastrom. Testovacími úlohami boli rôzne veľkosti sady grafov s ofarbovaním hrán.

Experimenty overenia riešenia v simulačnom prostredí CloudSim ukázali, že hypotéza H2 je vyvrátená. To znamená, že neplatí, že zelené počítanie v cloudových systémoch spomaľuje činnosť počítačového cloudu, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.

Na základe vykonania experimentálneho overovania konštatujeme, že pokiaľ nejde o mimoriadne veľké výpočtové úlohy, môžeme vyvrátiť hypotézu H1 - Cloudové systémy, na báze virtuálnych strojov, spomaľujú činnosť fyzických výpočtových prostriedkov, čo vedie k zhoršeniu kritérií kvality pri rozvrhovaní úloh v systéme.

Podľa výsledku skúmania dávkovania úloh v použitých prostrediach konštatujeme záver, že zamietame hypotézu H3 - Dekompozícia cloudovej aplikácie negatívne ovplyvňuje náklady na prevádzku výpočtového systému..

Do budúcnosti navrhujeme rozvoj zlepšenia danej problematiky, a to skúmaním vylepšenia predstavených vlastných algoritmov. Napriek úspešnému splneniu kritérií v simulovanom cloudovom prostredí, by mohli byť získané výsledky zlepšené ešte s menšou mierou porušenia SLA so zachovaním zníženej energetickej náročnosti. V reálnom cloudovom prostredí je priestor na vylepšenie užší, ale predstavuje väčšiu výzvu. Tú by mohol predstavovať zmenený prístup k spracovaniu testovacej úlohy a spôsobu plánovania v takomto prostredí.

Literatúra

- [1] J. Yang, C. Liu, Y. Shang, Z. Mao a J. Chen, „Workload predicting-based automatic scaling in service clouds”, in *Proceedings of the 2013 IEEE Sixth International Conference on Cloud Computing, ser. CLOUD '13*, júl 2013, s. 810-815. DOI: [10.1109/CLOUD.2013.146](https://doi.org/10.1109/CLOUD.2013.146).
- [2] P. Martincová, *Rozvrhovanie úloh v gride*, Žilina: Žilinská Univerzita v Žiline, habilitačná práca, 2008.
- [3] G. Moltó, M. Caballer, a C. de Alfonso, „Automatic memory-based vertical elasticity and oversubscription on cloud platforms“, in *Future Generation Computer Systems*, marec 2016, s. 1–10. DOI: [10.1016/j.future.2015.10.002](https://doi.org/10.1016/j.future.2015.10.002).
- [4] S. Palúch a Š. Peško, *Kvantitatívne metódy v logistike*, Žilina: Žilinská Univerzita v Žiline, 2006, s. 185, ISBN-80-8070-636-0.
- [5] G. Pfister, *High Performance Mass Storage and Parallel I/O*, IEEE: Press and Wiley Press. 2001, Kapitola 42, An Introduction to the InfiniBand Architecture, s. 617–632, ISBN: 978-0-471-20809-9.
- [6] R. Buyya a S. Venugopal, „A Gentle Introduction to Grid Computing and Technologies“, Computer society of India, [Online]. Available: <http://gridbus.cs.mu.oz.au/~raj/papers/GridIntro-CSI2005.pdf>. [Cit. Marec 2019].
- [7] I. Foster, „What is the Grid? A Three Point Checklist“, Argonne National Laboratory & University of Chicago, [Online]. Available: <http://www.mcs.anl.gov/~itf/Articles/WhatIsTheGrid.pdf>. [Cit. Január 2019].
- [8] J. Škrinárová, *Elastický klaster*, Banská Bystrica: Univerzita Mateja Bela, 2016, s. 15, ISBN: 978-80-557-0642-9.
- [9] B. Sivák a J. Škrinárová, „The Neural Networks in Quantum Computing“, In *Electronic Computers and Informatics ECI 2006: proceedings of the Seventh International Scientific Conference*, Košice: Faculty of Electrical Engineering and Informatics of the Technical University of Košice, 2006. s. 48-51, ISBN: 80-8073- 598-0.
- [10] J. Suchý, J. Škrinárová a P. Trhan, „Neural Network Modelling of Friction in Robot Joints“, in *Proceedings of the 8th International Symposium ISMCR 98*, Praha: ČTU, 1998, s. 157-162, ISBN: 8001018148.
- [11] J. Emeras, S. Varrette, V. Plugaru a P. Bouvry, „Amazon Elastic Compute Cloud (EC2) vs. In-House HPC Platform: A Cost Analysis“, *IEEE Transactions on Cloud Computing*, roč. 7, č. 2, s. 456-468, jún 2019. DOI: [10.1109/TCC.2016.2628371](https://doi.org/10.1109/TCC.2016.2628371).
- [12] A. Prabhakaran a J. Lakshmi, "Cost-Benefit Analysis of Public Clouds for Offloading In-House HPC Jobs", in *2018 IEEE 11th International Conference on Cloud Computing (CLOUD)*, júl 2018, s. 57-64. DOI: [10.1109/CLOUD.2018.00015](https://doi.org/10.1109/CLOUD.2018.00015).
- [13] Amazon AWS, „Amazon EC2 Instance Types“, [Online]. Available: <https://aws.amazon.com/ec2/instance-types/>. [Cit. November 2019].
- [14] T. Wood, P. Shenoy, A. Venkataramani a M. Yousif, „Black-box and gray-box strategies for virtual machine migration“, in *Proceedings of the 4th USENIX Conference on Networked Systems Design and Implementation*, apríl 2007, s. 1-17.

- [15] A. Beloglazov, „*Energy-Efficient Management of Virtual Machines in Data Centers for Cloud Computing*“, Department of Computing and Information Systems The University of Melbourne, [Online]. Available: <<https://beloglazov.info/thesis.pdf>>. [Cit. Apríl 2019].
- [16] J. Chabarek, J. Sommers, P. Barford, C. Estan a D.T.S. Wright, „Power awareness in network design and routing“, in *IEEE INFOCOM 2008 - The 27th Conference on Computer Communications*, máj 2008, s. 457–465. DOI: [10.1109/INFOCOM.2008.93](https://doi.org/10.1109/INFOCOM.2008.93).
- [17] S. K. Moghaddam, „*Anomaly-aware Management of Cloud Computing Resources*“, School of Computing and Information Systemaddam, School of Computing and Information Systems The University of Melbourne, [Online]. Available: <<http://www.cloudbus.org/students/SaraKardaniPhDThesis2019.pdf>>. [Cit. Október 2019].
- [18] K. Vaidyanathan a K. S. Trivedi, „A comprehensive model for software rejuvenation“, *Proceedings of the IEEE Transactions on Dependable and Secure Computing*, roč. 2, č. 2, s. 124–137, apr. 2005, ISSN:1545-5971. DOI: [10.1109/TDSC.2005.15](https://doi.org/10.1109/TDSC.2005.15).
- [19] A. Beloglazov a R. Buyya, „Managing overloaded hosts for dynamic consolidation of virtual machines in cloud data centers under quality of service constraints“, *IEEE Transactions on Parallel and Distributed Systems*, roč. 24, č. 7, s. 1366-1379, aug. 2012, ISSN: 1045-9219. DOI: [10.1109/TPDS.2012.240](https://doi.org/10.1109/TPDS.2012.240).
- [20] A. Beloglazov, J. Abawajy a R. Buyya, „Energy-Aware Resource Allocation Heuristics for Efficient Management of Data Centers for Cloud Computing“, in *Future Generation Computer Systems*, máj 2012, s. 755-768. DOI: [10.1016/j.future.2011.04.017](https://doi.org/10.1016/j.future.2011.04.017).
- [21] M. Bala a Devanand, „Performance evaluation of cloud datacenters using various green computing tactics“, in *2015 2nd International Conference on Computing for Sustainable Global Development (INDIACom)*, máj 2015, s. 956-961. url: <<http://www.sciencedirect.com/science/article/pii/S0167404811001672>>.
- [22] A. Beloglazov a R. Buyya, „*Optimal online deterministic algorithms and adaptive heuristics for energy and performance efficient dynamic consolidation of virtual machines in cloud data centers*“, *Concurrency and Computation: Practice and experience*, [Online]. Available: <<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.224.7146&rep=rep1&type=pdf>>. [Cit. August 2019].
- [23] L.A. Zadeh, „Fuzzy sets“, in *Information and Control*, roč. 8, č. 3, s. 338-353, jún 1965, DOI: [10.1016/S0019-9958\(65\)90241-X](https://doi.org/10.1016/S0019-9958(65)90241-X)
- [24] E.H.Mamdani a S.Assilian, „An experiment in linguistic synthesis with a fuzzy logic controller“, in *International Journal of Man-Machine Studies*, roč. 7, č. 1, s. 1-13, nov. 1973, DOI: [10.1016/S0020-7373\(75\)80002-2](https://doi.org/10.1016/S0020-7373(75)80002-2)
- [25] UK-MAC, „*CloverLeaf*“, [Online]. Available: <<https://uk-mac.github.io/CloverLeaf/>>. [Cit. Jún 2019].
- [26] E. Vesel, J. Škrinárová a A. Dudáš, „Comparison of in-house HPC calculation with public cloud computing for parallel algorithm containing recursive functions“, in *ICETA 2019: 17th International conference on emerging elearning technologies and applications : Information and communication technologies in learning*, nov. 2019, s. 805-809. ISBN 978-1-7281-4967-7.
- [27] House of Graphs - Snarks. [Online]. Available: <<https://hog.grinvin.org/Snarks>>. [Cit. Marec 2020].
- [28] Amazon AWS, „*AWS ParallelCluster*“, [Online]. Available: <<https://aws.amazon.com/hpc/parallelcluster/>>. [Cit. September 2019].

Zoznam publikovaných prác autora

- [1] ***Optimization design for parallel coloring of a set of graphs in the High-Performance Computing [print]*** / Dudáš Adam (40%) - Škrinárová Jarmila (40%) - Vesel Eduard (20%). In: IEEE 15th International Scientific Conference on Informatics [print] : proceedings. - 1. vyd. - New York: Institute of Electrical and Electronics Engineers, 2019. - ISBN 978-1-7281-3178-8. - s. 93-99 [print].
- [2] ***Comparison of in-house HPC calculation with public cloud computing for parallel algorithm containing recursive functions [electronic]*** / Vesel Eduard (40%) - Škrinárová Jarmila (40%) - Dudáš Adam (20%). In: ICETA 2019 [electronic] : 17th IEEE International conference on emerging eLearning technologies and applications : Information and communication technologies in learning. Starý Smokovec, Slovakia. November 21-22, 2019 : proceedings. - 1. vyd. - Denver: Institute of Electrical and Electronics Engineers, 2019. - ISBN 978-1-7281-4967-7. - s. 805-809 [online].
- [3] ***Model of education and training strategy for the management of HPC systems [electronic]*** / Škrinárová Jarmila (40%) - Dudáš Adam (40%) - Vesel Eduard (20%). In: Informatics 2017, roč. 14 [print, electronic] : IEEE International Scientific Conference on Informatics. - 1. vyd. - Danvers: Institute of Electrical and Electronics Engineers, 2017. - ISBN 978-1-5386-0888-3. - s. 400-405 [print].
- [4] ***Model of education and training strategy for the high performance computing /*** Škrinárová Jarmila (50%) - Vesel Eduard (50%). In: ICETA 2016 [elektronický zdroj] : 14th IEEE international conference on Emerging eLearning technologies and applications : November 24-25, 2016, Starý Smokovec, The High Tatras, Slovakia. - [S.l.]: IEEE, 2016. - ISBN 978-1-5090-4699-7. - CD-ROM, s. 315-320.
- [5] ***Teaching effectiveness of high-performance computing using an online course /*** Vesel Eduard (90%) - Škrinárová Jarmila (10%). In: International education and research Journal (IERJ) [elektronický zdroj] : peer reviewed international journal. - ISSN 2454-9916. - Vol. 2, no. 7 (2016), s. 96-97. [online].
- [6] ***Private cloud solution for an engineering company /*** Vesel Eduard (100%). In: Ad Alta : journal of interdisciplinary research. - ISSN 1804-7890. - Vol. 5, iss. 1 (2015), , s. 108-110.